

Alarma de Proximitate cu Jurnal Serial

Introducere

Proiectul constă în realizarea unui sistem de securitate inteligent, capabil să monitorizeze distanța până la obiectele din jur și să declanșeze o alertă complexă (vizuală, sonoră și digitală) în momentul încălcării unui perimetru de siguranță configurabil.

- **Ce face:** Sistemul utilizează un senzor ultrasonic pentru a măsura distanța în timp real. Utilizatorul poate regla „limita de alarmă” folosind un potențiomtru. Valorile sunt afișate pe un ecran LCD, iar în caz de proximitate critică, un buzzer pasiv și un LED emit semnale de avertizare. În paralel, sistemul trimite un jurnal de evenimente către un PC via Serial (USART).
- **Scopul lui:** Implementarea și integrarea a patru interfețe hardware fundamentale: **ADC** (citire potențiomtru), **I2C** (afișare LCD), **Timere/PWM** (măsurare distanță și control buzzer) și **USART** (comunicație cu PC-ul).
- **Ideea de pornire:** Nevoia de a crea un sistem de monitorizare interactiv, unde pragurile de declanșare nu sunt „hardcodate”, ci pot fi ajustate manual de utilizator, oferind în același timp o metodă de monitorizare la distanță prin jurnalul serial.
- **Utilitate:** Proiectul demonstrează conceptele de bază ale sistemelor de alarmă moderne și ale monitorizării industriale, fiind un punct de plecare excelent pentru aprofundarea protocoalelor de comunicație și a procesării semnalelor senzoriale.

Descriere generală

Arhitectura proiectului este centrată pe microcontrolerul **ATmega328P**. Sistemul este organizat în trei piloni funcționali:

Modulul de Achiziție și Control (Intrare)

- **Senzorul HC-SR04:** Emite ultrasunete și măsoară timpul de î ntoarcere a ecoului pentru a calcula distanța.
- **Potențiometrul:** Acționează ca un divizor de tensiune. Tensiunea variabilă este convertită de **ADC** într-o valoare numerică ce definește pragul de distanță setat de utilizator.

Modulul de Procesare

Microcontrolerul compară distanța citită de senzor cu pragul setat prin potențiomtru. Acesta gestionează temporizarea pentru senzorul ultrasonic și calculează frecvența pentru semnalul de alarmă.

Modulele de ieșire (Feedback)

- **Feedback Vizual (I2C):** Afișează pe LCD 1602 distanța curentă și pragul setat.
- **Feedback Sonor și Luminos (PWM):** Controlează un buzzer pasiv și un LED. Frecvența sunetului se modifică în funcție de cât de aproape este intrusul.
- **Jurnal Serial (USART):** Trimite mesaje de stare către terminalul PC-ului (ex: "Sistem activ", "ALERTA: Obiect la 15 cm").

Hardware Design

În această etapă, lista de componente hardware necesare pentru realizarea proiectului este următoarea:

Nr. Crt.	Componentă	Cantitate	Rol în proiect	Modul / Interfață folosită
1.	ATmega328P-Xplained Mini	1 buc.	Unitatea centrală de procesare.	Toate
2.	Senzor Ultrasonic HC-SR04	1 buc.	Măsurarea distanței prin ultrasunete.	Timere / Input Capture
3.	Display LCD 1602 + I2C	1 buc.	Afișarea distanței și a limitei setate.	I2C (SDA, SCL)
4.	Potențiomtru 10kΩ	1 buc.	Reglarea manuală a pragului de alarmă.	ADC
5.	Buzzer Pasiv	1 buc.	Alarmă sonoră cu ton variabil.	PWM / Timere
6.	LED 3mm (Roșu)	1 buc.	Alertă vizuală în caz de proximitate.	GPIO / PWM
7.	Rezistență 220Ω	1 buc.	Protecția LED-ului.	N/A
8.	Breadboard & Fire Dupont	1 set	Realizarea conexiunilor fizice.	N/A

Software Design

Descrierea codului aplicației (firmware)

Codul este conceput pentru a rula pe microcontrolerul ATmega328P, utilizând o abordare modulară pentru a facilita debugging-ul și extinderea ulterioară. Fiecare periferic (ADC, USART, I2C) are propriul set de funcții de inițializare și utilizare.

- **Mediu de dezvoltare:** Microchip Studio (fostul Atmel Studio 7) sau VS Code cu extensia PlatformIO.
- **Librării și surse 3rd-party:**
 - **I2C Master Library:** Pentru gestionarea comunicării pe bus-ul I2C.
 - **LCD I2C Library:** Adaptată pentru controller-ul PCF8574 (cel de pe spatele LCD-ului), pentru a

trimite comenzi și date text.

- **avr/io.h & avr/interrupt.h:** Librăriile standard pentru lucrul cu registrele și întreruperile AVR.

Algoritmi și structuri de date

• Algoritmul de măsurare a distanței:

- Se emite un puls de 10μs pe pinul de Trigger.
- Se utilizează un Timer (Timer1 pe 16 biți) pentru a măsura durata pulsului de ecou recepționat.
- Distanța este calculată prin formula: $distanța = (timp_ecou * 0.034) / 2$.

• Procesare ADC (Potențiometrul):

- Valoarea brută (0-1023) este mapată într-un interval util pentru alarmă (ex: 10 - 80 cm).
- Se folosește un sistem de mediere a ultimelor 5 citiri pentru a elimina jitter-ul (fluctuațiile mici ale valorii afișate).

• Logica de Alarmă (State Machine):

- **Starea SAFE:** Distanța reală > Pragul setat. LED-ul este stins, buzzer-ul este inactiv.
- **Starea WARNING:** Distanța scade sub prag. LED-ul clipește rar, buzzer-ul emite un sunet de joasă frecvență.
- **Starea CRITICAL:** Distanța < 10 cm. LED-ul stă aprins, buzzer-ul emite un sunet continuu/strident.

- **Protocol USART:** Datele sunt formate ca string-uri și trimise către PC la intervale de 1 secundă pentru monitorizare.

Surse și funcții implementate


Funcție	Rol
void USART_init(unsigned int baud)	Configurează rata de transfer și formatul frame-ului pentru comunicarea serială.
void ADC_init()	Configurează referința de tensiune și pre-scaler-ul pentru convertorul analog-digital.
uint16_t ADC_read(uint8_t channel)	Returnează valoarea digitală citită de la potențiometrul.
void T1_init()	Inițializează Timer1 pentru măsurarea precisă a timpului (modul Input Capture).
void PWM_init()	Configurează Timer0/2 pentru generarea semnalului PWM către buzzer.
void LCD_print(char* str)	Funcție helper pentru afișarea șirurilor de caractere pe ecranul I2C.
void process_alarm()	Funcție principală de decizie care compară datele și activează ieșirile.

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:
<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:
<http://ocw.cs.pub.ro/courses/pm/prj2026/atoader/marian.panait1206> 

Last update: **2026/05/14 23:26**